

# USE OF TRANSACTION AGENTS TO PERFORM DISTRIBUTED TRANSACTIONS

## BACKGROUND OF THE INVENTION

### Field of the Invention

**[0001]** The invention relates to the field of network computing. More specifically, the invention relates to using transaction agents to perform distributed transactions.

### Description of the Related Art

**[0002]** Providing services to “network elements” (e.g. Switches; Routers; Application Servers; File Servers; End-user Workstations; participants in a TDM ring using SONET/SDH BLSR or UPSR etc.) in a networking environment may consist of many tedious and burdensome tasks.

## CIRCUIT CREATION

### **[0003]** Overview

**[0004]** For example, one method of providing connectivity between “switching network elements” (e.g., routers; switches; gateway servers; participants as nodes of a TDM ring using SONET/SDH and BLSR or UPSR; etc.) is to build a network circuit connection. Network circuit connections provide dedicated connectivity between switching network elements, thereby insuring reliable network throughput and availability along the connection. The process of building a circuit between two or more switching network elements is called network provisioning. Some of the steps associated with provisioning include route selection, physical provisioning, and equipment interconnection. However, today’s heterogeneous networking environment makes establishing a circuit connection a tedious task, brimming with potential pitfalls, resulting in a high probability of network failures.

**[0005]** Figure 1 illustrates how remote circuit provisioning is used to build a circuit by transmitting provisioning instructions to remote switching network elements. In this scenario, a technician connects remotely to a first switching network element 120 from a Graphical User Interface (GUI) 110 running at a remote location. While remotely connected to the first switching network element 120, one may execute a set of instructions to the switching network element 120, instructing the switching network element to provision a cross connect from the first switching network element 120 to a second switching network element 130.

**[0006]** Thereafter, the technician would similarly connect to the remaining switching network elements (switching network element 140, switching network element 150, and switching network element 160), in turn, performing similar operations until a complete circuit connection is built across the network. Depending on the number of network elements, this may become a long process. Since the circuit connection is fully set up in a given switching network element before moving onto the next, what are referred to as partial circuits are created until the last switching network element is processed to complete the circuit.

**[0007]** Additional Detail

**[0008]** A more detailed description of circuit provisioning follows. The traditional circuit provisioning process generally begins with selecting a network route or path to have provisioned. Route selection may include examining a network topology map in order to view the network bandwidth. Here, a manual check may be performed at each switching network element to determine if network bandwidth is available to make the circuit connection feasible. If bandwidth is available, then each switching network element along the path is provisioned. If bandwidth is not available, then an alternate path is selected or new switching network element(s) deployed within the network to provide additional bandwidth.

**[0009]** Primarily, a switching network element routes data within a network and/or from one network connection to another network connection. When provisioning a circuit a “cross connect” is executed within a switching network element to direct an input from the input facility of a first network connection to an output facility of a second network connection. A facility is a set of channels within a switching network element used in connecting the two network connections.

**[0010]** Typically, to provision a switching network element from a remote location, a remote connection is first established with the remote switching network element. A message to disable a first necessary facility is sent to the switching network element. This is followed by a check for a network failure and a check to determine if the connection to the switching network element has failed (e.g. server down, command failed, etc.).

**[0011]** Subsequent to the confirmed disabling of the first facility, a message to disable a second necessary facility is sent to the switching network element and again a check for a network failure is performed. Thereupon, a check for a disconnect failure to the switching network element is again performed.

**[0012]** After both facilities are disabled, a cross connect message is sent to the switching network element to generate the cross connect between both facilities. Then, a check for a network failure is performed, followed by a check for a cross connect failure (server down, command failed, etc.).

**[0013]** Following the programming of the cross connect, a message is sent to enable the first facility. Thereupon, a check for a network failure and an enable failure are performed. If successful, a message is sent to enable the second facility. A check for a network failure is again performed and then a final check is done to determine if an enable failure occurred.

**[0014] Problems**

**[0015]** A mistake to the provisioning of the circuit connection (e.g. wrong determination of network bandwidth) causes problems. Additionally, a mistake may cause the connection between the remote location and one or more of the network elements to be broken. If so, the remote location cannot be used to undo what was done. Rather, the effected switching network element will need to be corrected on location. At a minimum, these mistakes may require the technician to undo at least some of the installed cross connects.

**[0016]** Problems also materialize when an error occurs while un-provisioning a circuit connection. Un-provisioning, which provides for the termination of circuit connections, suffers deficiencies similar to provisioning mainly because un-provisioning also requires the manipulation of each network element to complete the task. For example, if a network technician fails to remove a circuit cross connection from a network element, then that part of the circuit will appear to be in service on a network topology map when, in fact, it is available and has no traffic going across it. Therefore, during subsequent attempts to provision a connection the circuit will appear to be in use, causing one to select a different path when in fact the connection is available. Due to this inaccuracy, inefficiencies develop, causing bottlenecks and delays along certain network paths and potentially no network traffic along others.

**[0017]** Worse yet, if network bandwidth is indeed available but hidden, network managers may prematurely schedule network elements to be upgraded or overlaid, causing an additional inefficient use of resources in time and cost.

**[0018]** Even if a network technician correctly provisions or un-provisions the circuit connection, other similar problems may present themselves if the technician delays or completely foregoes updating the topology map to show the provisioning or un-provisioning of the circuit.

## SOFTWARE INSTALLATION/UPGRADE

[0019] Another example of the burdensome and overwhelming tasks of providing services in a distributed network environment includes installing/upgrading software on remote network elements. Typically, in a decentralized software environment each network element has its own individual copy of a software application resident upon its hard drive. When installing new software applications or software upgrades, software technicians must install in each individual network element the new applications or upgrades. Whether remotely or manually on location this scenario is problematic if the installation or upgrade of the software is an "all-or-nothing deal." For example, in a file-sharing environment, the upgraded software may not be compatible with the older version software. Here, if only some of the network elements upgraded receive the software, file sharing between upgraded and non-upgraded network elements may be hindered until all network elements have been upgraded or the upgraded network elements are restored to the original version. This scenario is especially unacceptable if file-sharing is essential to the user's business.

## **SUMMARY OF THE INVENTION**

**[0020]** Use of transaction agents to perform distributed transactions is described.

According to one embodiment of the invention a coordinator transaction agent that includes an itinerary and a state machine is instantiated. The itinerary indicates a plurality of network elements and the plurality of network elements is in communication with the coordinator transaction agent. The coordinator transaction agent causes itself to be replicated onto the plurality of network elements according to the itinerary. Each of the replicated transaction agents causing an indication of their replication to be communicated back to the coordinator transaction agent. The coordinator transaction agent coordinating operations of the state machine in each of the replicated transaction agents to implement a distributed state machine.

## **DESCRIPTION OF THE DRAWINGS**

**[0021]** The invention may best be understood by referring to the following description and accompanying drawings that are used to illustrate embodiments of the invention. In the drawings:

**[0022]** Figure 1 illustrates how remote circuit provisioning is used to build a circuit by transmitting provisioning instructions to remote switching network elements.

**[0023]** Figure 2 illustrates how distributed transaction agents are used to install cross connects along a single circuit path through interconnected switching network elements according to one embodiment.

**[0024]** Figure 3 illustrates the contents of an exemplary itinerary according to one embodiment.

## DETAILED DESCRIPTION OF THE INVENTION

[0025] In the following description, numerous specific details are set forth to provide a thorough understanding of the invention. However, it is understood that the invention may be practiced without these specific details. In other instances, well-known protocols, structures and techniques have not been shown in detail in order not to obscure the invention.

[0026] The techniques shown in the figures can be implemented using code and data stored and executed on network elements. Such network elements store and communicate (internally and with other network elements over a network) code and data using machine-readable media, such as magnetic disks; optical disks; random access memory; read only memory; flash memory devices; electrical, optical, acoustical or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc. Of course, one or more parts of the invention may be implemented using any combination of software, firmware, and/or hardware.

[0027] The invention provides an apparatus and method of using transaction agents to perform distributed transactions. Transaction agents may be used to perform any of a number of distributed transactions. For example, , distributed transaction agents can be used to provision a circuit connection over a set of switching network elements in a consistent and automated fashion, eliminating costly manual processes. As another example, transaction agents can be used to install/upgrade software to multiple network elements. It should be understood, however, that the invention is not limited to the exemplary uses of transaction agents identified herein.

[0028] Figure 2 illustrates how distributed transaction agents are used to install cross connects along a single circuit path through interconnected switching network elements according to one embodiment. By way of example, these switching network elements may be part of an optical fiber, TDM ring, such as a SONET UPSR or BLSR. Here, transaction

agents are replicated to four switching network elements in the path to provision cross connects, facilitating a complete circuit.

**[0029]** In general, a transaction agent is a type of state machine that may replicate itself among numerous switching network elements, and depending on specific conditions, perform autonomous operations or transactions in furtherance of a common goal. In one embodiment, the states of the distributed state machine are of two types, replication and task. The replication type involves the replication of the transaction agent to different network elements. In contrast, the task state(s) involve the actual work for which the transaction agent was created. In one embodiment described herein, there is a single replication state (during which the transaction agent is replicated on every network element involved) that is followed by a set of one or more task states. However, alternative embodiments could have additional replication states, and/or intermix replication states with task states. The transaction agents are notified when it is legal to commit a set of one or more transactions and transition to a different state.

**[0030]** Furthermore, each of the transaction agents can determine if the global operation should be aborted (e.g. based on notification, lack of communication, etc.). If the transaction agents abort their operation, a roll back function is invoked at each individual switching network element, thereby restoring the switching network elements to their original state. Hence, the transaction agents may roll back, leaving the network unaffected.

**[0031]** In one embodiment, an end-user (e.g., a network manager) uses a GUI 210 to initiate a process to determine the switching network elements to which to provision a circuit. This may take the form of using a network topology map 220 to determine a list of switching network elements for provisioning (itinerary list) and a list of the facilities in those switching network elements to cross connect (facility list). Figure 3 illustrates the contents of an exemplary itinerary according to one embodiment. In one embodiment, the itinerary is not an active component in that it does nothing without external stimulus.

**[Siegfried, is this illustration of an itinerary list adequate? How should I represent the facilities on this figure?]**

**[0032]** Thereafter an initial transaction agent is instantiated on switching network element 230. This switching network element also typically contains the element management system (EMS) which manages the network. In one embodiment, this initial transaction agent is labeled the coordinator transaction agent 232, while replications of this coordinator transaction agent are referred to as replicated transaction agents. The coordinator transaction agent 232 has stored therein the itinerary and facility list, and is responsible for coordinating the transition state changes of the replicated transaction agents.

**[0033]** The coordinator and replicated transaction agents communicate with each other through simple messages. In one embodiment, communications between network elements are handled through a Java based mobile agent environment. Agent environments are well known components of Java based Mobile Agent applications and generally hosts the agents and facilitate mechanisms for inter-agent communication.

**[0034]** After initialization, a coordinator transaction agent 232 is given to an agent environment 234 that invokes the replicating of the transaction agent as per the itinerary. Here, replication to each of the switching network elements specified in the itinerary occurs when the coordinator transaction agent's 232 communication thread is first initiated within the agent environment 234. In one embodiment, this replication occurs in a "worm" like manner. In another, each replication is generated by the coordinator transaction agent.

**[0035]** In figure 2, a replicated transaction agent 242 arrives at switching network element 240. Responsive to being placed in the agent environment 244 on the switching network element 240 the first state transition is invoked. Here, the replicated transaction agent causes a check to see if it can access the provisioning Application Programming Interface's (API's) of the switching network element 240. Provisioning APIs are the specific hardware and/or software commands used by each individual switching network element to provision a cross connection. The replicated transaction agent 242 attempts to access the

provisioning APIs by requesting the local agent environment 244 for a handle to the API. For security reasons, the agent environment 244 knows the location and privileges of the replicated transaction agent 242 and can determine if it should permit this reference. When the replicated transaction agent 242 determines it has authentication to execute all the transactions (by way of the APIs) available to generate the required cross connection, the replicated transaction agent returns with a "success" control token (assuming of course that it was able to get the API reference). If replication is in a "worm" like manner, the agent environment 244 then causes the replicated transaction agent 242 to replicate itself onto the next switching network element in order of the itinerary, here switching network element 250, and perform similar logic to this switching network element in turn.

**[0036]** In one embodiment, when the last switching network element 270 in the itinerary is reached, the transaction agent environment 274 returns a state control token back to the coordinator transaction agent 232. In another embodiment, the control token may travel back, in reverse order, from the last participant replicated transaction agent to the coordinator transaction agent 232. Regardless, of the method, at this point in time, the coordinator transaction agent 232 exists on the switching network element (230), and one replicated transaction agent exists on each of the switching network element 240, 250, 260, and 270 (replicated transaction agent 242, replicated transaction agent 252, replicated transaction agent 262, and replicated transaction agent 272).

**[0037]** If all the replicated transaction agents have successfully returned a "success" state control token with regard to the replication, the replication state of the distributed state machine has been completed.

**[0038]** In one embodiment, the one or more subsequent task state transitions invoke the replicated transaction agents to perform circuit provisioning transactions. Such as to disable the two facilities, enter the cross connect, and then re-enable both facilities. In another embodiment, the replicated transaction agent provisioning transactions may further include disabling the first facility, checking for a disable failure, disabling the second facility,

check for a disable failure, entering a cross connect, checking for a *cross connect failure*, enabling the first facility, checking for an enable failure for the first facility, enabling the second facility, and lastly, perform the last checking for an enable failure for the second facility.

**[0039]** When using transaction agents to provision circuits, the task states can be implemented in a variety of ways. For example, there could be a single task state in which the cross-connects are fully installed. Alternatively, there could be multiple task states, in each of which part of the provisioning is performed in each of the switching network elements involved. For example, in one embodiment, the replicated transaction agents transition from one state to another when they successfully disable all facilities on the switching network element. Next, the replicated transaction agents transition to another state upon successfully entering the cross connect on their switching network element. The replicated transaction agents transition to yet another state upon successfully enabling all facilities within the switching network element.

**[0040]** In one embodiment, the replicated transaction agents transition from state to state in lock step with each other. That is, the replicated transaction agents perform the provisioning transactions in parallel or simultaneously. In another embodiment, each of the replicated transaction agents performs one state transition, in turn, before the first transaction agent is allowed to transition to its next state. For example, in one embodiment, responsive to the completion of the replication state(s) of the distributed state machine, the coordinator agent 232 sends a new state token to the replicated transaction agent 242. Upon receiving this token from the coordinator transaction agent 232, the agent environment 244 invokes the next state transition for the replicated transaction agent 242. This state invokes the replicated transaction agent 242 to request from the agent environment 244 the identity of the current switching network element. Using this value, the replicated transaction agent 242 determines which facilities are to be disabled by associating its facility list with switching network element 240. Upon successfully disabling

all facilities for switching network element 240, the state returns with success, and the agent environment 244 forwards the next state token to the next switching network element in the itinerary. This procedure repeats itself until all replicated transaction agents have completed their designated transactions. Of course, it is to be understood the invention is not limited to the described embodiments for transitioning states and alternative embodiments may be implemented that remain within the scope of the invention.

**[0041]** As a result of the autonomous nature of the replicated transaction agents, the coordinator transaction agent 232 need only coordinate the replicated transaction agents replication and state transitions on the switching network elements; the coordinator transaction agent 232 does not have to check for network failures. Rather, each of the replicated transaction agents check for network failures in their respective switching network element. The coordinator transaction agent 232 waits to receive status messages from the replicated transaction agents.

**[0042]** If a problem develops during the replication or the state of machine operation, the roll back feature is invoked. Such roll back features can be responsive to a number of stimuli including a communication from another transaction agent, a temporal constraint, etc.

**[0043]** For example, if one of the replicated transaction agents fails to perform an operation, the particular state control token returns with a descriptive failure message. In one embodiment a failure state may be initiated by operator control as well as by one of the state machines' failure/abortion. If a failure occurs, rather than issuing a success state control token, the replicated transaction agent issues a failure state control token to all replicated transaction agents in the itinerary. These replicated transaction agents then proceed to invoke an exception state. The exception state causes the roll back -- each replicated transaction agent autonomously undoes/reverses whatever state transitions have occurred.

[0044] Where temporal constraints are also used, the exception state is invoked upon a time-out condition when a switching network element fails to successfully perform its required set of transactions within a specific period of time. For example, assume the transaction agents have been replicated to half of the switching network elements (e.g., switching network element 240 and switching network element 250) when network communication fails between the switching network elements in the network. Here, the replicated transaction agent 262 is unable to notify the others that its transactions have either succeeded or failed. In this situation, the replicated transaction agent 242 and replicated transaction agent 252 on the switching network element 240 and switching network element 250 respectively will eventually time-out. Therefore, these replicated transaction agents will autonomously roll back their state transitions within a specific time frame. To provide another example, assume that replication of the transaction agent to the required switching network elements has been successful, but that the performance of one of the transactions by one of the replicated transaction agents has caused a communication failure (e.g., a failure in the communication link between the EMS in the switching network element 230 and one or more of the switching network elements 240, 250, 260, and 270; a failure in communication between different ones of the switching network elements; etc.). Previously, this type of failure would often require a technician to go to the physical location of the network element to undo the transaction(s) so that communication could be reestablished. However, as a result of this communication failure, communication between at least certain of the replicated transaction agents (referred to herein as the "isolated replicated transaction agent") and the coordinator transaction agent will not occur. The implementation of the roll back feature triggered by a temporal constraint (a timeout) will result in these isolated replicated transaction agents undoing their transactions (those replicated transaction agents that are not isolated, will also implement the roll back function as a result of a temporal constraint and/or a failure message), thereby reestablishing the communication link between the various switching network elements.

Thus, this type of automatic transaction roll back eliminates the need for a technician to go to the remote location in those situations where the provisioning transactions has caused connections between the network management system server and a network element to fail.

**[0045]** Thus, in one embodiment, upon a failed transaction, the coordinator transaction agent 232 simply reports the failure to the user who invoked the process; it does not need to do any further processing, because it knows that all replicated transaction agents have rolled back their state, as per the defined exception handler. Furthermore, it is understood that the replicated transaction agents will terminate automatically upon completion of the exception state.

**[0046]** In one embodiment, the coordinated execution of distributed transactions across a network is performed using a modified two phase commit (2PC) protocol. Typically, the first phase of the protocol completes when the coordinator transaction agent 232 receives a state control token from all replicated transaction agents. In one embodiment, the success state control token indicates to the coordinator transaction agent 232 that each replicated transaction agent is prepared to enter the second phase and commit or execute a set of circuit provision transactions on each switching network elements. That is, all the replicated transaction agents are prepared to execute a set of provision APIs and transition to a next state. A fail state control token will invoke the coordinator transaction agent 232 to provide the requestor with a message describing the processes failure.

**[0047]** For example, a success state control token received by the coordinator transaction agent 232 on the last executed operation is treated as being equivalent to a “yes” vote for all prior distributed transactions. If at any point prior to this a transaction failed to execute on any replicated transaction agent, a “no” vote is assumed. The second phase of the protocol involves sending out a commit (in the case of a yes vote) or roll back message (in the case of a no vote) to each of the replicated transaction agents.

**[0048]** In one embodiment, a receive and forward strategy is used to communicate messages to the replicated transaction agents. That is, the coordinator transaction agent 232 initiates the distributed transactions by sending a message to the replicated transaction agent 242 on the first switching network element in the itinerary list. That replicated transaction agent and each subsequent replicated transaction agent is then responsible for forwarding any received messages to its neighbor (e.g., the next switching network element as specified by the itinerary). Each agent will only report transition to a success state if all previous agents in the itinerary have also reported transition to a success state. Therefore, in this embodiment, status messages received by the coordinator transaction agent are cumulative. This minimizes any communication bottlenecks resulting from a relatively slow-speed link .

**[0049]** In an alternative embodiment, a centralized control methodology is used to communicate the messages. Here, the coordinator transaction agent 232 receives status messages from and forwards all instructions to each replicated transaction agent. That is, the coordinator transaction agent 232 acts as a centralized control. For example, the coordinator transaction agent 232 will replicate a transaction agent to the switching network element 240 as described above, however, upon completion of replication to the switching network element 240, the coordinator transaction agent 232 will connect directly to the next switching network element 250 on the itinerary list and await a response that the second replicated transaction agent 252 has replicated. This process continues until all the replicated transaction agents have arrived at the switching network elements specified on the itinerary list. In this embodiment, no communication is performed between the replicated transaction agents and thus inter-node communication is between each individual replicated transaction agent and the coordinator transaction agent.

**[0050]** In certain embodiments, distributed transaction agents are used to un-provision circuits. That is, distribute transactions are replicated to remote switching network elements to tear-down a circuit. Here, the coordinator transaction agent is created

containing an itinerary list of the switching network elements and the cross connection to be disabled. The coordinator transaction agent then replicates the transaction agents to the switching network elements by way of the various embodiments described to provision a circuit. However, here, the replicated transaction agents perform transactions to un-provision the cross connections.

**[0051]** Another exemplary use of transaction agents is to label connections with identifiers. For instance, in a network environment containing cross connects, transaction agents may be replicated to each switching network element with transactions to tag each cross connection with an identifier. Upon completion, a message will be returned from the replicated transaction agents to the coordinator transaction agent 232 containing the identifier of each cross connection. The cross connection identifiers may be used to view, edit, and delete connections after they have been created. In another embodiment, this information may be provided to applications such as a network Topology Map, Alarm/Events, Root Cause Analysis, and Alarm Correlation applications for associating faults in the network with user resources.

**[0052]** Another exemplary use of transaction agents is the distribution of software to multiple network elements. Here, the distributed transaction agents perform a set of transactions to install or upgrade software on network elements provided in an itinerary. In one embodiment, the installation or upgrade of software on network elements must be performed on all the network elements on the itinerary or else the installation or upgrade is useless (e.g., an "all or nothing" scenario). In this embodiment, if all of the replicated transactions agents successfully installed or upgraded the software, all transactions will be committed. In contrast, if any of the replicated transactions agents fail to install or upgrade the software, all transactions will be rolled back.

**[0053]** It should be understood that although an embodiment distinguishes between the coordinator transaction agent and replicated transaction agents, the coordinator transaction agent may perform the functionality's of a replicated transaction agent.

**[0054]** While the invention has been described in terms of several embodiments, those skilled in the art will recognize that the invention is not limited to the embodiments described.

**[0055]** The method and apparatus of the invention can be practiced with modification and alteration within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of limiting on the invention.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219